Software Requirements Specification

for

Analisis Tren Film Berdasarkan Data IMDb

**Version 1.0 approved**

**Authored by:**

1. **Adelina Vivian Magdiel (103052300059)**
2. **Felicia Cyntia Febriani (103052300086)**
3. **Nydia Artha Carissa (103052300101)**
4. **Keisha Hernantya Zahra (103052330063)**
5. **Sarah Aisyah (103052300022)**

**Telkom University**

**21/03/2025**

# Table of Contents

[**Table of Contents**](#_bogd4orcbeed)

[**Revision History**](#_32ecmhdps2u7)

[**Team’s Role**](#_tn1fgmd0b2bi)

[**1. Introduction**](#_elgc9ujzzqmo)

[1.1 Purpose](#_g0i9wfdyvmqm)

[1.2 Document Conventions](#_css8zgnfwk13)

[1.3 Intended Audience and Reading Suggestions](#_kvw59ce7xloq)

[1.4 Project Scope](#_dhg5abksn5xq)

[1.5 References](#_83dcm642rhxj)

[**2. Overall Description**](#_di19frt6lsfd)

[2.1 Product Perspective](#_ptl4hga2zgs6)

[2.2 Product Features](#_ylisdsyw713)

[2.3 User Classes and Characteristics](#_px2ffse8q3um)

[2.4 Operating Environment](#_ehvisfezldub)

[2.5 Design and Implementation Constraints](#_rodjxe29pdhu)

[2.6 User Documentation](#_izd6sa3fhxg0)

[2.7 Assumptions and Dependencies](#_9qo5cbag5a1l)

[**3. System Features**](#_2w0dd916z4op)

[3.1 Data Collection & Processing](#_de6n9hbz70b)

[3.2 Visualization of Movie Trends](#_mo4d159qjw21)

[3.3 User Interaction & Filtering](#_wtr6uut0wces)

**4. Test Scenario**

4.1 The requisites

4.2 Test Scenario

# Revision History

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Team’s Role | 11 April 2025 | Penambahan pembagian tugas dan peran anggota tim dalam dokumen | 1.1 |
|  |  |  |  |

# 

# Team’s Role

| Nama | Role | Tugas |
| --- | --- | --- |
| Adelina Vivian Magdiel | Frontend Developer & UI/UX | * Mendesain tampilan web menggunakan Figma * Mengimplementasikan HTML & CSS * Membuat dashboard yang responsif dan user-friendly |
| Felicia Cyntia Febriani | Backend Developer | * Integrasi dengan IMDb API * Mengatur proses pengambilan dan penyimpanan data * Menyusun struktur backend dengan Flask/Streamlit |
| Sarah Aisyah | Filtering & User Interaction | * Membuat fitur pencarian dan filter film * Menjamin waktu respon cepat (< 3 detik) * Menangani interaksi pengguna (misalnya klik film) |
| Nydia Artha Carissa | Data Processing | * Membersihkan dan mengolah data dari IMDb * Menyiapkan data untuk visualisasi * Menggunakan Pandas untuk analisis statistik dasar |
| Keisha Hernantya Zahra | Data Visualization | * Membuat grafik interaktif (rating, genre, tren) * Menggunakan Matplotlib, Seaborn, atau Plotly * Menghubungkan visualisasi ke data filter |

# Introduction

## Purpose

Aplikasi ini bertujuan untuk menganalisis tren film berdasarkan data dari IMDb, memungkinkan pengguna untuk mengeksplorasi informasi mengenai rating, genre, dan popularitas film dalam berbagai periode waktu. Aplikasi ini membantu pengguna dalam memahami pola dan preferensi film secara lebih mendalam melalui visualisasi data yang interaktif.

## Document Conventions

* Semua requirement dalam dokumen ini diberi label REQ-# untuk identifikasi yang unik.
* Prioritas fitur ditandai dengan High, Medium, atau Low berdasarkan tingkat kepentingan.
* Semua referensi ke API atau pustaka eksternal ditulis dalam format italic.
* Istilah teknis didefinisikan dalam Appendix A: Glossary.

## Intended Audience and Reading Suggestions

Dokumen ini ditujukan untuk:

* Developer → memahami kebutuhan sistem dan mengimplementasikan teknis
* Data Analyst → menggunakan aplikasi untuk analisis data tren film
* Peneliti → menganalisis tren film untuk studi akademik atau penelitian pasar
* Pengguna umum → mengeksplorasi informasi mengenai film favorit mereka secara interaktif

## Project Scope

Aplikasi ini akan menyediakan:

* Pengambilan data film dari IMDb API.
* Analisis statistik mengenai rating, genre, dan tren popularitas film.dalam kurun waktu 1 bulan dan 1 tahun terakhir.
* Visualisasi interaktif seperti grafik rating film, popularitas genre, dan tren penayangan.
* Fitur filtering dan eksplorasi berdasarkan kategori tertentu berdasarkan genre film
* Rekomendasi film untuk ditonton pengguna berdasarkan interaksi pengguna di aplikasi

## References

* IMDb API Documentation
* Pandas, Matplotlib, Seaborn untuk analisis dan visualisasi data
* Python Flask/Streamlit untuk antarmuka pengguna

# Overall Description

## Product Perspective

Aplikasi ini merupakan alat analitik yang memungkinkan pengguna untuk melihat tren film dari waktu ke waktu, dengan fokus pada rating, genre, dan popularitas.

## Product Features

* Mengambil data film dari IMDb API.
* Menganalisis tren film menggunakan metode statistik dasar.
* Menyediakan visualisasi interaktif dalam bentuk grafik dan diagram.
* Mendukung filtering dan pencarian film berdasarkan tahun, genre, atau rating*.*

## User Classes and Characteristics

* Developer: memahami kebutuhan sistem dan mengimplementasikan solusi teknis
* Data Analyst: menggunakan aplikasi untuk mengeksplorasi dan memahami tren film berdasarkan dataset yang tersedia
* Peneliti: menggunakan aplikasi untuk mengumpulkan data dan menganalisis pola tren industri film
* Pengguna Umum: membutuhkan tampilan antarmuka yang intuitif dan mudah digunakan
* Produser Film & Cinephile: Menilai potensi genre atau tren yang sedang berkembang.

## Operating Environment

* Backend: Python (Flask/Streamlit untuk pemrosesan data dan visualisasi).
* Frontend: Dashboard interaktif berbasis web menggunakan bantuan HTML.
* Database: SQLite atau CSV untuk menyimpan hasil analisis sementara.
* UI/UX: Menggunakan desain responsif dengan pendekatan minimalis untuk kemudahan navigasi pengguna dengan bantuan Figma dan CSS.

## Design and Implementation Constraints

* Aplikasi harus tetap ringan agar bisa berjalan di perangkat dengan spesifikasi rendah.
* Bergantung pada IMDb API, sehingga perubahan atau keterbatasan API dapat memengaruhi fungsionalitas aplikasi.
* Database SQLite atau CSV digunakan untuk menyederhanakan penyimpanan data tanpa memerlukan server database tambahan.
* Flask/Streamlit digunakan untuk menjaga arsitektur tetap sederhana dan mudah dipahami oleh pengembang pemula.

## User Documentation

Dokumentasi pengguna yang akan disertakan dalam perangkat lunak ini meliputi:

* User Manual: Panduan langkah demi langkah dalam penggunaan aplikasi, termasuk cara mencari tren film dan menyaring data.
* Tutorial: Petunjuk berbasis video atau PDF yang menjelaskan bagaimana menggunakan fitur utama aplikasi.
* Format Delivery: Semua dokumentasi akan tersedia dalam format PDF dan HTML untuk kemudahan akses.

## Assumptions and Dependencies

* API IMDb harus tersedia dan dapat diakses.
* Koneksi internet diperlukan untuk pengambilan data real-time.
* Aplikasi dapat berjalan di perangkat desktop dan mobile browser.

# System Features

## Data Collection & Processing

3.1.1 Description and Priority

Fitur ini bertanggung jawab untuk mengambil dan memproses data film dari IMDb API. Data ini akan digunakan sebagai dasar untuk analisis tren film.

**Prioritas: High**

**Komponen Prioritas:**

**Benefit:** 9 (Penting untuk analisis data yang akurat)

**Penalty:** 7 (Tanpa fitur ini, data tidak bisa diperbarui)

**Cost:** 5 (Memerlukan koneksi API dan penyimpanan database)

**Risk:** 6 (Ketergantungan pada IMDb API dan kemungkinan perubahan skema data)

Fitur ini bertanggung jawab untuk mengambil dan memproses data film dari IMDb API. Data ini akan digunakan sebagai dasar untuk analisis tren film. Prioritas: High

3.1.2 Stimulus/Response Sequences

1) Pengguna menjalankan aplikasi.

2) Sistem menghubungi IMDb API dan mengambil data film terbaru.

3) Data di format ulang dan disimpan dalam database SQLite/CSV.

4) Sistem menampilkan konfirmasi bahwa data berhasil diperbarui.

3.1.3 Functional Requirements

1. REQ-1: Sistem harus dapat mengambil data film dari IMDb API.
2. REQ-2: Sistem harus membersihkan dan menyimpan data dalam format yang dapat dianalisis.
3. REQ-3: Jika koneksi API gagal, sistem harus memberikan pesan kesalahan kepada pengguna.

## Visualization of Movie Trends

3.2.1 Description and Priority

Fitur ini menyediakan visualisasi tren film dalam bentuk grafik interaktif untuk memudahkan pengguna memahami data.

**Prioritas: High**

**Komponen Prioritas:**

**Benefit: 8** (Memudahkan pengguna dalam memahami tren film)

**Penalty: 6** (Tanpa visualisasi, data sulit dianalisis secara intuitif)

**Cost: 5** (Memerlukan pustaka visualisasi tambahan seperti Matplotlib atau Seaborn)

**Risk: 4** (Kemungkinan keterbatasan performa jika dataset terlalu besar)

Fitur ini menyediakan visualisasi tren film dalam bentuk grafik interaktif untuk memudahkan pengguna memahami data. Prioritas: High

3.2.2 Stimulus/Response Sequences

1. Pengguna memilih parameter (tahun, genre, rating).
2. Sistem mengambil data yang relevan dan membuat grafik.
3. Grafik ditampilkan kepada pengguna dalam dashboard interaktif.

3.2.3 Functional Requirements

1. REQ-4: Sistem harus dapat menampilkan grafik tren popularitas film.
2. REQ-5: Sistem harus mendukung visualisasi heatmap untuk genre populer.
3. REQ-6: Sistem harus menampilkan distribusi rating film dalam diagram batang.

## User Interaction & Filtering

3.3.1 Description and Priority

Fitur ini memungkinkan pengguna untuk menelusuri dan memfilter data berdasarkan kriteria tertentu.

**Prioritas: Medium**

**Komponen Prioritas:**

**Benefit:** 7 (Meningkatkan pengalaman pengguna dalam eksplorasi data)

**Penalty:** 5 (Tanpa fitur ini, pengguna kesulitan menemukan film spesifik)

**Cost:** 4 (Memerlukan pemrosesan tambahan untuk filtering data)

**Risk:** 3 (Kemungkinan keterlambatan jika data sangat besar)

Fitur ini memungkinkan pengguna untuk menelusuri dan memfilter data berdasarkan kriteria tertentu. Prioritas: Medium

3.3.2 Stimulus/Response Sequences

1. Pengguna memasukkan filter (judul, genre, rating, tahun).
2. Sistem menampilkan hasil sesuai kriteria yang dipilih.
3. Pengguna dapat mengeklik film tertentu untuk melihat detail lebih lanjut.

3.3.3 Functional Requirements

1. REQ-7: Sistem harus memiliki fitur pencarian film berdasarkan judul.
2. REQ-8: Sistem harus memungkinkan pengguna memfilter film berdasarkan genre, tahun, dan rating.
3. REQ-9: Sistem harus merespons pencarian dengan menampilkan hasil dalam waktu kurang dari 3 detik.

# 4. Test scenario

**4.1 The requisites**

To start the test execution the following matters need to be present:

* Aplikasi web berbasis Python Flask/Streamlit
* Koneksi ke IMDb API aktif dan stabil
* Data preprocessing script berjalan lancar dan tanpa error
* Modul visualisasi aktif dan bisa dirender di semua resolusi
* Filtering dan pencarian responsif dan akurat
* Autentikasi API Key sudah terkonfigurasi
* Log error aktif untuk membantu debugging saat testing

Minimal system requirements:

* Browser: Chrome/Edge terbaru
* RAM: 4GB
* Koneksi internet stabil
* Resolusi layar minimal: 1366x768

System configuration:

* Backend: Python 3.x, Flask/Streamlit
* Frontend: HTML/CSS, Bootstrap (optional)
* Visualisasi: Matplotlib/Seaborn/Plotly
* Database: SQLite/PostgreSQL (opsional untuk skala besar)
* Library tambahan: Pandas, Requests, Scikit-learn (untuk ML), IMDbPy/API wrapper

Product risks (from MTP):

1. Kegagalan mengambil data dari IMDb API
2. Grafik gagal dimuat karena dataset yang besar
3. Filtering lambat pada dataset besar
4. Error saat input user tidak valid
5. Server backend tidak merespons
6. Crash saat mengambil data secara bersamaan dari banyak pengguna
7. Rekomendasi film tidak relevan dengan mood input user

**4.2 Test Scenario**

Link test case: [PASD \_test case](https://docs.google.com/spreadsheets/d/1eCkhzTMjqrYgUoUcOf8cHhBHgOgofmFBlSk2_UaTDY8/edit?usp=sharing)

Link hasil test: [hasil dari PASD \_test case](https://docs.google.com/spreadsheets/d/1jL36Y0rEI8T9KVSkSbAYAknqLpP1gBSDf99OABRrYfU/edit?usp=sharing)